【V4.8】企业版CodeGeeX使用手册

欢迎打开企业版CodeGeeX使用手册!

在这份为企业用户定制的CodeGeeX使用支持中,我们提供了简介、安装、客户端功能、 用户后台、使用案例、FAQ、反馈 七个模块,以方便您更好地使用这一强大的开发工具,希 望对您有所帮助。

1. 简介

CodeGeeX是一款基于大模型的AI智能编程助手,它可以实现代码的生成与补全,为代码 添加注释,不同编程语言的代码互译以及针对技术和代码的智能问答等功能,帮助开发者 显著提高工作效率。CodeGeeX支持300+种编程语言,适配多种主流IDE平台,包括 VSCode、JetBrains全家桶等。

企业版CodeGeeX亮点:

- 包含所有基础功能(代码生成、智能问答、单元测试、代码翻译、代码注释等)
- 支持交互式编程、检索增强生成(RAG)
- 基于企业自有的业务代码进行模型微调
- 支持本地私有化部署,提供企业级管理后台支持

支持的主流编程语言(30种):主流编程语言支持的效果更好。

分类	支持的语言			
前端	HTML、JSON、JavaScript、TypeScript、Vue、XML			
后端	C、C++、C#、Dart、Go、Haskell、Java、Kotlin、Lua、Objective- C++、Pascal、PHP、Python、R、Ruby、Rust、Scala、Shell、SQL			
APP	C#、Dart、Java、Kotlin、Swift			
其他	CUDA、Markdown、Tex、Txt			

请在附录中查看支持的所有编程语言(310种)

2.客户端(插件)安装

2.1 获取用户账号

1. 用户可从管理员获取 用户后台 的访问地址,该访问地址也是 Host地址,或者从网站的 横幅中复制。以下为网站横幅示例:

● 客户端Host地址为用户后台页面根地址: http://10.50.220.87 □,复制该地址至客户端中完成Host配置。 式配置 我知道了 点击地址旁的复制icon、并粘贴至客户端设置页面中的host地址

- 2. 通过网页浏览器,用户可以从网页进行登录,从而获取操作手册和下载插件。有以下几 种方式可以获得登录账号:
 - 如果 用户后台 已开放注册,用户可以通过用户后台登录页面的注册按钮进行用户注册,完成后可使用注册的账号进行登录。

	注册
欢迎登录管理后台	名 请输入账号
☑ 请输入账号或邮箱	各 请输入姓名
▲ 请输入密码	☑ 请输入邮箱地址
□ 记住密码	▲ 密码 🖉
登录	▲ 请二次确认密码
注册	注册
	返回登录
进入用户后台界面,选择注册	注册用户信息

。 如果 用户后台 未开放注册, 用户可以通过管理员获取用户账号进行用户登录。

• 如果 用户后台 已对接SSO或其他企业内部账号,则通过点击登录页面的 企业登录 完成用户登录。

2.2 CodeGeeX for Visual Studio Code 插件

运行环境

该插件基于 CodeGeeX 代码生成模型,适配 Visual Studio Code 集成开发环境(1.72.0版本及以上),支持Windows、MacOS、主流Linux桌面版、麒麟、统信等操作系统。稳定版本的信息请咨询管理员。

下载地址

登录后可以在下载与支持菜单中,客户端下载页面获取VS Code的插件。兼容版支持VS Code 1.72.0 - 1.77.0版本。

88 / 下载与支持 / 客户端下载	
客户端信息	
×0 VS	SCode 🔶 JetBrains
在Visual Stud	dio Code中安装CodeGeeX企业版
标准版: 下載器件 亲	兼容版 (适配VSCode 1.72-1.77): 下载50件
步骤—: 1.在1开相似栏的软件市场后,在右上角的更多造项中,选择从vsx安装。 2.之后选择下驾突的插件进行变装。	步端二: 1.通作奖装完成后,打开插件市场,找到安装的CodeGlee公企业版插件。 2.点击右上角的更多,打开设置页面,在插件设置页面中完成Hout的能面。 3.配置另Hout后,在插件中完成量录。
Image: state in the state	

安装步骤

- 1. 在 Visual Studio Code 中点击左侧 Extensions(扩展)按钮。
- 2. 点击 Extensions 侧边栏右上角的 ... ,打开菜单。
- 3. 点击菜单中的 Install from VSIX ... 。
- 4. 选中本地的安装包文件,并点击 Install 安装。
- 5. 安装后重新启动 IDE 即安装成功。

	个人或篇 ()) 三 · 圖 · ● package Q 圖室 ④ 晶式化用 名用 名用 名用 ○ Codespaces 6.004 Get / p 8.1 MB 2/P / B 今天 20.53 ○ 下信 CodeGet / p 8.1 MB 2/P / B 今天 20.53 ○ 大術 A.2 MB 文術 今天 20.53 9.7 / 20.53 ○ Codeget / p 6.1 MB 2/P / B 今天 20.53 9.7 / 20.53 ○ Codeget / p 6.1 MB 2/P / B 9.7 / 20.53 9.7 / 20.53 ○ Codeget / p 6.1 MB 2/P / B 9.7 / 20.53 9.7 / 20.53 ○ Codeget / p 6.1 MB 2/P / B 9.7 / 20.53 9.7 / 20.53 ○ Codeget / p 6.1 MB / B 9.7 / 20.53 9.7 / 20.53 ○ Codeget / p 6.1 MB / B 9.7 / 20.53 9.7 / 20.53 ○ Codeget / p 6.1 MB / B 9.7 / 20.53 9.7 / 20.53 ○ Codeget / p 6.1 MB / B 9.7 / 20.53 9.7 / 20.53 ○ Codeget / p 7.8 / 20.53 9.7 / 20.53 9.7 / 20.53 ○ Codeget / p 7.8 / 20.53 9.7 / 20.53 9.7 / 20.53 ○ Codeget / p 7.8 / 20.53
Batter Gts Syntax	• Red 新建文件夹 取消 安装

配置说明

企业版 CodeGeeX 需配置服务器地址后才能使用。配置方法:

- 1. 侧边栏点击CodeGeeX的图标打开CodeGeeX侧边栏,点击右上角更多设置。
- 2. 找到 Codegeex: Host 设置项,输入管理后台的访问地址,地址示例: http://192.168.0.1:2080。

Ð	CODEGEEX	≓设置 ×	1 D	
ρ	Ask CodeGeeX 代码翻译 补全溯源 回	@ext:AMiner.codegeex-ent	erprise 找到12个设置 云 5	
20	◎ 设置	用户	备份和同步设	n
-4	名 个人中心	> 扩展(12)	Codegeex: Host	
æ	CodeGeeX 欢迎使用企业 ② 帮助		当您向我们的赛型发送请求时,您可以将其配置为控制域名或IP地址。	
₿	2 请输入账号或邮箱	1. 找到设置	http://10.50.220.87/	
\$ \$>	诸统人肤号或部箱	1	Codegeex.Chat: Language Preference	
a	☆ 请输入密码 Ø	0.308911	为聊天回复设置语言。 Default	
	✓ 记住密码 网页登录	2. 设直HOSt		
	登录		Codegeex.Comment: Language Preference 许择你相差估计器语言	
			Coderen Old by U. Langer P. Grand	
			Codegeex.statebar Or: Language Preference 设置您的例边栏UI语言	
			Default V	
			Codegeex: Candidate Num	
			每次补全的最多选项数目,选项越多,生成速度趋慢。	
			1(例) ~	
0			▼ 10430年2011年20日、 10日2月20日日元2018月21日、 11日2月1日日11日日日11日日11日日11日日11日日日11日日日11日日	
8			Codegeex: Completion Delay	
£63			生成進送到時(初,用素避光不必要到建议。	

点击侧边栏后点击设置,设置Host

设置说明:

设置项	设置内容	说明
Host	填写服务器地址	通过配置Host,可以控 制向模型发送请求时的

		域名或IP地址
Language Preference 语言	English 中文 Default	设置聊天回复、代码注 释、插件侧边栏UI的语 言
Candidate Num 候选数量	1(fast) —— default 2(medium) 3(slow)	设置候选代码的数量; 候选数量越多,生成补 全建议花费的时间越 长。
Code Repository Infiling 代码仓库级补全	勾选,默认不选中	在管理后台完成配置的 情况下,是否参考当前 的代码项目生成补全建 议。
Completion Delay 补全延迟	填写数字(单位为秒),默 认0.5,最低0.1	指在编辑器中无操作时 开始获取代码补全建议 前的延迟时间。可以调 整此值,以便在使用 CodeGeeX扩展进行编 码时获得更佳体验,同 时减少不必要的补全建 议。
Disabled For 禁用语言	填写item(key)和value	CodeGeeX临时禁用的 特定语言列表,可以手 动设置语言作为键,然 后将值设置为true以禁 用某种语言,或设置为 false以重新启用它(需 要重新启动VSCode)。
Enable Extension 启用扩展	勾选,默认不选中	如果您想启用 CodeGeeX的隐身模 式,请勾选此选项。
Generation Preference 生成偏好	Automatic—— default Line by line Block	选择生成偏好。 • automatic:根据输 入生成代码行或代码 行。

		 Line by line,每次生成一行代码和/或一行注释。 Block:每次生成多行(代码块)。
Only Key Control 仅按键控制	勾选,默认不选中	如果您希望在隐身模式 下仅在需要时通过按下 Alt/Option + \ 键 获得建议,请勾选此选 项。
Thrid Party Function 三方功能调用	勾选,默认不选中	在管理后台配置完三方 的授权信息后,允许Al 调用第三方接口的功 能,例如Gitlab、Gitee 等。
Command Display Mode 指令展示模式	勾选,默认选中	勾选后,将系统默认指 令展示在代码文件内每 个函数上方。
Command Display Type 指令展示类型	Text —— default 文字 Icon 图标	勾选指令展示模式后, 通过选择指令展示类 型,选在系统默认指令 在函数上方的展现方 式。文字类型为平铺展 示、图标类型为收起的 列表。

登录使用

企业版CodeGeeX 插件需要登录才可以使用全部功能。登录方法如下:

- 点击左侧的 CodeGeeX 图标,打开 CodeGeeX 侧边栏,在侧边栏中输入邮箱和密码进 行登录,登录成功后即可开始使用。
- 或点击 网页登录 跳转至浏览器进行登录,在浏览器中登录成功后,回到 VS Code 即可 开始使用。(点击VS Code中的头像选择账号后可以登出)

Ð	CODEGEEX	≠ 设置 ×	<u>۳</u>	
ρ	Ask CodeGeeX 代码翻译 补全溯源 〇		prise 환희 12 수 문법	
, 		用户	备份和同	步设置
••••••••••••••••••••••••••••••••••••••		→ 扩展 (12)	Codegeex: Host 当您有我们协模思发送课来时,您可以将其配置为控制结合或P地址。 http://t0.50.220.87/ Codegeex.Chat: Language Preference 力博天回复设置语言。	
			Codegeex Comment: Language Preference 法律位结果的注释语言, 中文 ~ Codegeex.Bidebar UI: Language Preference 设置包约例如近年/11音 Default ~	
0			Codegeex: Candidate Num 每次升全的量多过调数目,迅速超多,生成速度趋像。 1 (快)	
3 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3	20.40 Min		Codegeex: Completion Delay 生成延迟时间(秒),用来避免不必要的建议。 c =	() 0

点击侧边栏后输入邮箱和密码登录

升级更新

CodeGeeX 插件支持在IDE内进行升级更新。更新方法如下:

- 点击 企业版CodeGeeX 侧边栏中的更多选项中的下拉框,选择 升级 。
- 或点击右下角升级弹窗中的 升级 。

CODECEEX Ask CodeGeeX Pro 代码翻译 项目地图 补全其 ○ ① 第二步 ⑫ 谋 ④ 个人中心 ② ⑦ 帮助 第一步 欢迎使用 CodeGeeX ⑦ 升级 ⑦ ⑦ ⑦ 第一步 欢迎使用 CodeGeeX ⑦ 升级 ⑦ <td< th=""><th>+ ~ [2 zsh]] ・ (2 zsh]] ・ (1) ・ (1) ・ (2)</th></td<>	+ ~ [2 zsh]] ・ (2 zsh]] ・ (1) ・ (1) ・ (2)
通过侧边栏进行升级	通过升级弹窗进行升级

2.3 CodeGeeX for JetBrains IDEs 插件

运行环境

该插件基于CodeGeeX 代码生成模型,适配 JetBrains 公司的系列 IDE 环境(2021.3版本 及以上),支持Windows、macOS、主流Linux桌面版、麒麟、统信等操作系统。稳定版本 的信息请咨询管理员。

适配的 IDE 包括 IntelliJ IDEA、PyCharm、WebStorm、Android Studio、GoLand、 PhpStorm、CLion、Rider、DataGrip、DataSpell、RubyMine、AppCode 及 Aqua 等。

下载地址

登录后可以在下载与支持菜单中,客户端下载页面获取JetBrains系列的插件。标准版可支持多种JetBrains的IDEs;兼容版支持IntelliJ IDEA 2022.1.3之前的版本。

器 / 下载与支持 / 客户编下载							
客户端信息							
	▲ VSCode ● JetBrains 在Jetbrains中安装CodeGeox企业级,示例为Intellij IDEA						
步彟一: 1.依次启击Intelli IDEA中的File, Settings和Plugins, 2.点击品件市场右侧的虚轮, 选择从睡量安装插序。 3.从硬盘选择完编件后,点击Install进行安装,安装完成后重启ID		参编二: 1.重启印6后,从其孝贞面选择Sottings。 2.在SetEngy得跟中,我们的这就中PCOdeGeeX,之后在CodeGeeX的设置页面完成Host地址的起置。 3.页词beet提着EdEI的中方成点是。					
A - S - S - S - S - S - S - S - S - S -	Ale vandeling Ale ale ale ale ale ale ale ale ale ale a	Image: Second					
Average A	net net Net	Constraints and a second a se					

安装步骤

- 1. 在IDE菜单中找到Settings项并点击,打开设置窗口。
- 2. 在设置窗口左侧栏中点击 Plugins ,并点击顶部的设置图标。
- 3. 点击菜单中的 Install Plugin from Disk ,选择本地的安装包文件并打开。安装后 重启 IDE 即可使用。

			and the second sec		
	Q.	Plugins		2	
intellij IDEA File Edit View Naviga	> Appearance & Behavior			- 75	
About Intellij IDEA	Keymap > Editor				
Check for Updates	Plugins Plugins Version Control				Settings
Settings # ,	Build, Execution, Deployment Languages & Frameworks Toole	Jump to Line			Marketplace Installed 🗢
Services >	Settings Sync Advanced Settings	Aust			Manage Plugin Repositories Programmin HTTP Proxy Settings
Hide IntelliJ IDEA %H 2		AND Yould		10	Show all Scala Manage Plugin Certificates
Hide Others 70%H 3		14.0M (0 3.01		1.5	Install JetBrains s. Install Plugin from Disk
Show All		Grazie Professional			Install Disable All Downloaded Plugins Enable All Downloaded Plugins
Quit IntelliJ IDEA %Q		A Solarized Theme			Install Overview What's New Reviews Additional Info
8		File Watchers			
9		JetBrains Academy			
	(2)			Cancel Apply OK	

打开设置窗口	打开插件设置	从本地安装

配置说明

企业版 CodeGeeX 需配置私有化部署的服务器地址后才能使用。配置方法:

1. 侧边栏点击CodeGeeX的图标打开CodeGeeX侧边栏,点击右上角更多设置。

2. 在 Host 选项中输入管理后台的访问地址,地址示例: http://192.168.0.1:2080 。

3. 点击右下 OK 。

Carl Control Cont	Seath Daryshell Coulds 0 de N Ma 300 Read Mar 31 Heighton 12 / P Dep Ten Hein 13 point nn	Converties 0 0 4 0 0 Converties 0 0 0 0 0 0 Converties 0 0 0 0 0 Converties 0	e marine Para Gana Jan	Contract & Market and Market	The Second Secon		
找到Settings			找到	Cod	egeex: Host		

设置说明:

设置项	设置内容	说明
Host 域名	填写服务器地址	通过配置Host,可以控 制向模型发送请求时的 域名或IP地址
Candidate Num 候选项个数	1(fast) — default 2(medium) 3(slow)	设置候选代码的数量; 候选数量越多,生成补 全建议花费的时间越 长。
Only Key Control 只用快捷键	勾选,默认不选中	勾选后会停用自动补 全,只能通过按下 Alt/Option + \ 键 触发AI生成补全建议。

Third Party Function 调用第三方功能	勾选,默认不选中	在管理后台配置完三方 的授权信息后,允许Al 调用第三方接口的功 能,例如Gitlab、Gitee 等。
Code Repository Infilling 代码仓库级补全	勾选,默认不选中	在管理后台完成配置的 情况下,是否参考当前 的代码项目生成补全建 议。
Generation Preference 生成偏好	Automatic—— default Line by line Block	选择生成偏好。 • automatic:根据输 入生成代码行或代码 行。 • Line by line,每次生 成一行代码和/或一行 注释。 • Block:每次生成多 行(代码块)。
CodeGeeX UI 侧边栏语言 Comment language 代码注释语言 Chat language 聊天回复语言	Default English 中文	设置插件侧边栏UI、代 码注释、聊天回复的语 言
Command Display Mode 指令展示模式	勾选,默认选中	勾选后,将系统默认指 令展示在代码文件内每 个函数上方。
Command Display Type 指令展示类型	Text —— default 文字 Icon 图标	勾选指令展示模式后, 可选择指令展示类型, 选在系统默认指令在函 数上方的展现方式。文 字类型为平铺展示、图 标类型为收起的列表。

Completion Delay 补全延迟	填写数字(单位为秒),默 认0.5,最低0.1	指在编辑器中无操作时 开始获取代码补全建议 前的延迟时间。可以调 整此值,以便在使用 CodeGeeX扩展进行编 码时获得更佳体验,同 时减少不必要的补全建 议。
Disabled For 禁用的语言	文本输入框	CodeGeeX临时禁用的 特定语言列表,手动填 写语言名称并使用英文 逗号分割,例如 java,python ,设置 后将禁用某种语言的补 全,删除后则重新启 用。

登录使用

- 点击右侧的 CodeGeeX 图标,打开 CodeGeeX 侧边栏,在侧边栏中输入邮箱和密码进 行登录。
- 或点击 网页登录 跳转至浏览器进行登录,在浏览器中登录成功后,回到 IDE 中即可开始使用。(点击IDE右下角CodeGeeX的logo选择账号后可以登出)



点击侧边栏后输入邮箱和密码登录

升级更新

CodeGeeX 插件支持在IDE内进行升级更新。更新方法如下:

- 点击 企业版CodeGeeX 侧边栏中的更多选项中的下拉框,选择 升级 。
- 或点击右下角升级弹窗中的 升级 。

 Ask CodeGeeX Pro 代码翻译 补全溯源 C C ② 第二步 第二步 ○ 设置 第二步 ○ 介人中心 ③ 帮助 次迎使用 CodeGeeX ① 帮助 第三步 ① 升級 第三步 ○ 介級 ① 新助 第三步 ○ 介級 ○ 新助 ○ 第助 ○ 第 ○ 第 ○ 和助 ○ 第 ○ 第 ○ 和助 ○ 第 ○ 和助 ○ 第 ○ 和助 ○ 和助<	 CodeGeeX Enterprise 系统检测到您当前的插件版本较旧,有些功能可能无法 使用,请下载最新版本的插件进行升级。 升级 3日内不再提醒
通过侧边栏进行升级	通过升级弹窗进行升级

冲突提示

 当插件安装后,有时候会有弹窗提示JetBrains自带的补全被禁用,忽视此提示即可,不 影响插件正常使用。



3. 客户端功能

企业版CodeGeeX客户端功能主要包含代码生成和补全、智能问答、企业知识库问答、单 元测试、代码翻译、代码注释、交互式编程、第三方功能调用等。

3.1 代码生成

CodeGeeX可以根据自然语言注释描述的功能自动生成代码,也可以根据已有的代码自动 生成后续代码,补全当前行或生成后续若干行,帮助提高编程效率。

当插件处于激活状态下,在编辑器中输入停止时,会从当前光标处开始进行自动的代码生成与补全。此时,右下角 CodeGeeX 图标转圈表示正在生成。生成的代码以灰色显示,按 Tab 即可将生成结果在当前位置插入。如果不需要使用生成的代码,可按 Esc 取消,也可以忽略生成的代码并继续输入。

按块/按行采纳: 当AI生成补全建议时,可以通过 Command+右 (Mac)或 Ctrl+右 (Windows) 按块采纳单行的部分代码;也可以通过 Command+下 (Mac)或 Ctrl+下 (Windows) 逐行采纳代码建议。

在配置中可以设置自动生成的方式:

- Automatic: 模型自主判断生成的行数
- Line by line: 逐行生成
- Block: 多行生成

以下是使用 JavaScript 时示例:

使用功能	示例
• 自动生成:在正常的编码工作流程中,稍微停下 光标,会看到CodeGeeX根据上下文给出的代码 建议。	<pre>33 ///// ******************************</pre>

注释生成:通过写一段注释,CodeGeeX可以根据注释含义提供相应的代码建议。注释按照正常注释规则,用简单准确的语言描述功能即可。



3.2 智能问答

问答模式

用户可通过与 CodeGeeX 聊天,直接询问问题并接受答案。对话模块可以回答与技术相关的各种问题,例如解释代码、编程语法、最佳实践、测试用例等。

- 使用对话功能,可以从 VS code 的左侧侧边栏点击 CodeGeeX 图标,并进入 Ask 标签。在对话模式中,输入问题并发送,即可得到模型回答。在输入问题的同时,同时在右侧编辑器选中代码,便可以结合选中的代码进行提问。
 注: Intellij IDEA 的侧边栏在 IDE 的右侧。
- 如果配置了三方模型,支持在问答场景中切换三方模型。
- 针对模型给出的回答,可以快速进行复制、插入、对比、运行等操作。
- 如果对模型给出的回答不满意,可以在回答的左下角点击 重新生成 的图标,让模型对 之前的问题重新给出回答。



快捷指令

使用功能 示例

- 使用 / 处理与特定命令相关的问题:
 - /explain: 解读选中的代码。如果没有选择代码,则默认解释全部代码。
 - /comment:为选中的代码添加逐行注释。
 如果没有选择代码,则默认为所有代码添加
 注释。
 - /fixbug:修复选中代码中的bug。
 - /tests : 为选中的代码生成单元测试。
 - /optimize : 为选中的代码提供优化建议。
- 在管理员通过提示词完成配置后,可以调用自定 义指令。



快捷指令的多种调用方式

调用方式	示例			
	CodeGeeX	>	Ask CodeGeeX	~ 光
• 右键菜单:	转到定义	F12	Explain Code	τŵΡ
	转到引用	企 F12	Fix Bug	νċγ
0 米牢侯式只 古 風 协 石 健 米	快速查看	>	Generate Comment	νŵγ
单在编辑器中调用。	本状的方门田	7- 0 540	Generate Tests	τφτ
	直找所有方用	L °0° F12	Optimize Code	Νŵγ

- 函数顶部(VSCode标准 版):
 - 点击函数顶部的指令进行 调用。
 - ・ 在设置中支持切换文字模 式和图标类型。
 - 在设置中支持打开或关闭 该功能。
- 函数左侧 (JetBrains):
 - 点击函数左侧图标进行调用
 - 在设置中支持打开或关闭 该功能。
- 支持的编程语言:
 - VSCode标准版:支持 Java、Go、Python、C、 C++、C#、Javascript、 Typescript、PHP
 - JetBrains支持: Java、
 Python、Go、Kotlin





基于代码仓库的智能问答

针对私有代码仓库的智能问答,使用了 RAG 检索增强生成的技术。因而对企业代码仓库中的知识,能在不进行模型微调的情况下,通过向量化数据的方式被有效检索,并在结果生成中进行增强,大幅减轻生成内容的幻觉。

另外,利用 @repo 的代码仓库进行智能问答时,针对代码片段的生成也更准确,因为模型 针对代码仓库进行检索,增强了生成能力;同时也会根据返回的代码溯源信息,找到代码 出处进行验证,实现更精准的代码生成。



示例:当管理员在管理后台完成企业代码仓库的配置后,用户可以通过 @repo 中选择管理 员配置的企业代码库进行智能问答。

选中企业代码库提问	基于斫洗代码库诵讨RAG生成回答
命令 ′/, 引用@ 调用 # ๋ <u>↑</u>	syslog/syslog.go#L28-L55 master/logrus_test.go#L743-L759
codegeex-4 V	master/logrus.go#L93-L114 master/logrus_test.go#L761-L769
@repo	查看引用 凶
	4. logrus_test.go:测试了LogLevel在不同的日志级别设置下的行为。
ms upload·2024-12-26	3. syslog.go:实现了将LogLevel映射到syslog日志级别的功能。
	2. logrus_test.go:测试了LogLevel的功能。
∕ cubox1	1. logrus.go:定义了LogLevel的枚举类型和相关的函数。
demo-2024-12-28	一 任 切
• equ-an-optorprise-edition/file-enline-proview	7. TraceLevel:追踪级别,用于记录程序运行过程中的更详细信息,通常只
cgx-qa-enterprise-edition/file-online-preview master-2024-12-26	调试过程中启用。
	6. DebugLevel:调试级别,用于记录程序运行过程中的详细信息,通常只在
₼ 114444	5. InfoLevel:信息级别,用于记录程序运行过程中的正常信息。
upload-2024-12-26	4. WarnLevel:警告级别,用于记录可能对程序功能广生影响但不会导致程 序无法正常运行的错误信息
A me	
codegeex-enterprise/apiserver main-2025-03-05	3. ErrorLevel:错误级别,用于记录导致程序功能缺失或无法正常运行的错
	记录此级别的日志时,程序会立即停止运行并打印错误信息。
codegeex-enterprise/code-auth-test	2. FatalLevel:严重级别,用于记录导致程序无法继续运行的错误信息。当
Main-2025-03-05	别的日志时,程序会立即停止运行并打印错误信息。
	在项目sillupsell/logids中,Logicever有以下/b种· 1. Panicl evel: 最高级别,用于记录导致程序崩溃的错误信息,当记录此级
main-2025-03-05	东西日sirupsop//aggus中,Lagleve/方以下口种:
	CodeGeeX 🗊 复制
代码仓库	

基于文档库的智能问答

针对企业文档库的智能问答,使用了 RAG 检索增强生成的技术。因而对项目的文档信息,可以通过向量化数据的方式被有效检索,基于文档库内容生成的答案深度融合了企业内部的研发文档知识,减轻了模型生成的幻觉。

当管理员在管理后台完成企业文档库的配置后,用户可以通过 @db 中选择管理员配置的企 业文档库进行智能问答。



基于本地文件的智能问答

针对本地文件的智能问答,允许用户上传本地代码文件和本地技术文档,并基于上传的本 地文件生成回答。通过本地文件问答功能,用户能够随时对本地文件进行研发问答,省去 了繁琐的步骤。

- 全級解題中, CodeGeeX会自动起出补全提示, 按TubI環境 + 送中代码 并急低, 可且行通知注释成代码部等等的 - → 使用 ^ + X + 1 向CodeGeeX世份	1 Condexed	AK COOKGENX (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)
	上传本地文件提问	基于上传的文件进行 回答

支持上传的文件类型

# 代码	马文件类型				
[&quo # 文≭	ot;html", ^{当类型}	"htm",	"json",	, "js",	&quo
" 🔨 [&quo	ot;doc",	"docx",	"pdf",	"txt",	&quo ⁻

3.3 单元测试

自动生成单元测试,既可以通过智能问答的交互方式实现,也可以通过内置的"/tests"命令实现,对于Java、Go、Python、C、C++、C#、Javascript、 Typescript、PHP等主流语言可以点击函数顶部的"生成单测 / Generate Tests"快捷指令。以下是实际编程工作中,使用CodeGeex的单元测试功能的步骤和注意事项:

- 1. 直接点击函数顶部的快捷指令,即可自动生成对应函数的单元测试代码。
- 2. 在你的代码中选择需要进行单元测试的特定函数或模块,可以是新增的功能函数,已修 复的bug或者经常容易出错的代码部分;选中代码后,点击鼠标右键选择CodeGeeX的

生成单测选项,或在侧边栏的问答框中输入"/tests",即可自动生成单元测试代码。



输入生成单元测试指令

Tips:添加注释有助于单元测试代码的准确性在选定的代码段上方,编写简单清晰的注释,描述函数功能或预期输出。CodeGeeX在生成测试代码时会参考这些注释的内容,注释越明确,CodeGeeX生成测试代码越精准。

如果你的函数涉及多个边界条件或特殊情况,可以在注释中明确指出,便于生成更全面的测试用例。

3. 在代码项目的目录栏中右键点击待测试文件,可以选择 CodeGeeX生成单测 ,从而针对 整个文件生成单元测试。文件级单测生成支持的编程语言为Java、Python、Go。

표정학원	JuserControllerjawa M Second	 ○ MailServiceImpLjava文件約年元割位文件正在生成中、清酌心等 後:0% 決測 CodeGeeX Enterprise 	▲ MailServicalmpLjava文件約単元測試文件 ② × MailServicalmpTest.java已经生成,请此击下方按钮意看生成 約単元測试 実選: CodeGeot Enterprise 更種 変種
点击生成了	文件级单测	单测文件生成中	完成单测文件生成

 在用户后台的个人页面可以指定几种主流语言的单元测试框架,便于根据指定框架生成 单元测试。

8 / 功能设置 / 单元测试						
单元测试生成						
▼ Java						
单元测试框架	默认	^				
 Python 	默认					
单元测试框架	JUnit4					
▼ Go	JUnit5					
单元测试框架	默认	~				
 JavaScript/TypeSet 	cript					
单元测试框架	默认	~				
▼ C/C++						
单元测试框架	默认	~				
		_				

设置单元测试框架

3.4 代码翻译

CodeGeeX 的代码翻译功能可以将一段写好的代码翻译为不同的编程语言。

1. 在CodeGeeX的侧边栏, 点击 代码翻译 或 Translation 。

2. 此时在编辑器中自由选择代码片段,该片段将自动出现在"Input Code"框中。



打开代码翻译

在翻译界面,选择目标语言后,点击翻译进行翻译;结果将显示在"输出代码"框中,点击插入可将翻译结果插入到编辑器中。



3.5 代码注释

在代码编辑区域,使用 右键菜单-CodeGeeX-生成注释 功能,可以解释代码含义并为代码增加注释。

注释生成支持中文和英文;在初次生成时,IDE会提示选择语言;之后可在插件设置中更 改。



添加过注释的代码

3.6 交互式编程(仅Visual Studio Code的标准版插件支持)

交互式编程InLine Chat是一种通过在代码中嵌入对话,与智能编程助手CodeGeex进行交 互的方式,可以使用"Command + I (mac) 或者 Ctrl + I (windows)"的快捷键触发 InLine Chat功能。

在代码上下文中提出问题,编程助手根据你的描述生成代码,或者直接为代码添加注释, 提升编程效率。

<pre>18 y_pred = model.predict(X) 19 20 plt.plot(x, y_pred, '-') 21 22 plt.show() 23 Start discussion // elase describe the code you want to generate.(e.g. Please help me write an Express server) </pre>	
19 plt.plot(x, y_pred, '-') 21 plt.show() 23 p	
20 plt.plot(x, y_pred, '-') 21 plt.show() 23	
21 22 plt.show() 23 Start discussion Please describe the code you want to generate.(e.g. Please help me write an Express server)	
22 plt.show() 23 p Start discussion Please describe the code you want to generate.(e.g. Please help me write an Express server)	
23 m Start discussion Please describe the code you want to generate.(e.g. Please help me write an Express server)	
Start discussion Please describe the code you want to generate.(e.g. Please help me write an Express server;	
Start discussion Please describe the code you want to generate.(e.g. Please help me write an Express server)	
Please describe the code you want to generate.(e.g. Please help me write an Express server)	

3.7 代码仓库级补全

在客户端的设置页面中,可以通过勾选Code Repository Infilling增强补全建议。通过参考 当前项目工程中的代码文件,模型可以生成更符合当前代码仓库的补全建议。

注意:当前代码仓库需要管理员在管理后台的企业知识库中完成对应代码仓库的配置,或 者用户在用户后台的个人知识库中完成个人代码仓库的配置。



3.8 第三方功能调用

在客户端的设置页面中,可以通过勾选Third Party Function调用第三方功能。第三方的配置需要在用户后台完成填写,在客户端的对话框中唤起"调用#"来调用创建合并请求、创建项目议题。

注意: 该功能需要在用户后台完成第三方平台的配置, 当前支持Gitlab和Gitee。如果切换 了三方模型, 该功能将不可用。



 Gitlab的凭证权限:需要勾选api、read_api、read_repository、write_repository、 read_registry和write_registry权限。

Uwenchang Gluwenchang	pur Stirlings / Access Takens ∰——∯7 Search settings				Personal Access Tokens Add a personal access token Token name For example, the application using the token or the purpose of the token.
Set status Edit profile Preferences	Personal Access Tokens Voi can generate a personal access token for each application you use that needs access to the GitLab APL ITTP. They are the only accepted password when you have Two-Factor Authentication (IFA) enabled. Action proceed by the the only accepted password when you have the factor authentication (IFA) enabled.	You can also use personal access 第四述	okens to authenticate agi	inst Git over	Expiration date 2024-09-22 Select scopes Scopes set the enrinision levels granted to the token. Learn more.
Content Conte	Totan name Scopes function-call epi, read_user, read_repository, write_repository, read_repidity, write_repidity teat-totan api, read_user, read_repository, read_repidity Teat-totan api, read_user, read_repidity Teat-totan api, read_user, read_user, read_repidity Teat-totan api, read_user, read_us	Created Last Use y Sep 05, 2024 2 months Nov 28, 2024 5 days at Nov 20, 2024 5 days at Nov 01, 2024 1 hour ag	Explese ago In 8 months all In 10 months all In 10 months calendar, It is visible in t this token.	Action	pl orant scomplete read/write access to the API, including all groups and projects, the container registry, and crast, api c
	找到Gitlab配置页面	5			配置Gitlab凭证

• Gitee的凭证权限:需要勾选Pull Request和Issue等权限。

dougliu1111	松大文h年可以使用私人受情访问 Gitee Open API	
-	测试 user info projects pull requests issues groups enterprises	東景錄 「《蜂游 使用私人令牌,可以通过 Gitee Open API 访问惊擾权的数据,请注意数据
消息中心	3个月前	第三步
鼻 我的消息		请填写的私人令牌描述,最大长度为50字符
*9 通知设置		
基本设置		请选择将要生成的私人令牌所拥有的权限
四 核导管理		□ 王述 ☑ user_info 访问你的个人信息、最新动态等
島 个人资料		projects 查看、创建、更新你的项目 了 pull requests 查看、发布。更新你的 Pull Request
8 新植物理		☑ issues
四 我的部递地址		notes 宣看、发布、管理你在项目、代码片线甲的评论 keys 宣看、部署、删除你的公钥
✔ 代码风格	第二步	hook 宣看、即署、更新你的 Webhook 亦至 他即形式外组织 [1] [1] [1] [2] [2]
ゆ 合用首造項		gitts 查看、删除、更新你的代码片段
(如今·12)要		enterprises 查看、管理你的企业以及成员 emails 演看你的个人即箱信息
A CHAN		
A 00001		観波
A 5.100		
gitee.com 👳		

3.9 新增代码行数上报

客户端支持统计用户当前的代码行数变动上报至管理后台,以便管理后台完成相关效能指 标的统计。

 如果用户希望只统计指定项目中指定代码文件或目录的代码行数变动,则需要在当前 IDE的代码项目中创建.codegeexcount配置文件,并在.codegeexcount配置文件中配置 需要统计的代码文件或目录。通过问答窗口创建的配置文件会提供默认的统计规则示 例,用户可以根据自己的需求进行修改。

Ask CodeGeeX 代码翻译 补全溯源 ① ① ——
<>>
欢迎使用 CodeGeeX
→ 在编辑器中, CodeGeeX会自动给出补全提示, 按Tab键采纳
→ 选中代码并点击右键,可进行添加注释或代码翻译等操作
→ 使用 ^ +
你可以试试这么问: Python中怎样遍历一个字典? 写一个快速排序函数。 有哪些给JavaScript的初学者的建议? 输入你的问题,或用'/'调用快捷命令 令'/. 引用@ 调用 #
点击创建.codegeexcount文件

- # 使用正则表达式构建上报规则,作为统计白名单,用于统计新增代码。
- # 1. 当项目中有符合该文件内规则的文件时,自动上报对应文件内的新增代码行数。
- # 2. 当该文件内没有可用规则时,则不进行新增代码行数的上报。
- # 3. 如果该文件不存在,则通过.gitignore文件内配置的忽略规则,忽略对应文件后,上报其余文
- # 4. 如果该文件和.gitignore均不存在,则不统计项目内所有文件的新增代码行数。
- 如果当前IDE的代码项目中不存在.codegeexcount文件,而存在.gitignore文件,则会统 计除.gitignore相关规则以外项目中的其他代码文件。
- 如果当前IDE的代码项目中既不存在.codegeexcount文件,也不存在.gitignore文件,则 不会统计项目中的所有代码文件。

3.10 代码审查和提交信息生成

在提交代码时,支持通过分支对变动的代码进行代码审查,并生成符合规范的提交信息。



代码审查



提交信息生成

3.11 项目地图(仅Pro版支持)

在客户端的设置页面中,可以通过勾选 Project Map 开启项目地图功能,从而在侧边栏 问答窗口的 项目地图 页面展示该项目的结构信息。通过参考当前的代码项目结构生成UML 图,开发者无需深入阅读每个功能点的代码细节,即可实现对项目整体架构的快速理解; 同时,帮助开发者分析并可视化各个模块之间的关系、依赖和交互,便于梳理复杂的模块 结构。

- 查看下一层:点击项目地图中的节点,通过 查看下一层 按钮,可以查看该节点对应的下 一层代码结构。
- 层级导航:通过左上方的下拉框,可以根据访问路径自由回到不同的层级,如果间隔超过5级,则再次点击 查看下一层 时会重新生成。
- 操作列表:通过右上方的操作列表,可以重新生成当前层级的UML图,同时支持以PNG 格式导出当前视图,或复制当前视图的图片源码。
- 视图模式: 通过右上方的切换视图, 支持切换图片模式和源码模式。
- 自由缩放:在底部支持通过放大镜按钮自由缩放。

注意:当前代码仓库需要管理员在管理后台的企业知识库中完成对应代码仓库分支的配置,或者用户在用户后台的个人知识库中完成个人代码仓库中该项目分支的配置。



4. 用户后台

企业版CodeGeeX的用户后台主要包含工作台、智能问答、工具箱、功能配置等主要功 能。

4.1 工作台

在用户后台中,支持用户通过工作台页面查看个人用户的代码采纳情况及相关的统计数 据。



4.2 网页版智能问答

- 在用户后台中,支持用户通过网页端访问研发相关的智能问答功能,包括调用命令、引 用代码库问答、引用文档库问答、上传本地文件问答等。具体细节请参考3.2。
- 2. 如果在问答中生成代码片段或文件,可以直接通过点击回答右下角的"运行XXX"按钮进 行效果预览。当前仅支持预览前端代码片段/文件。
- 3. 对于AI生成的回答,用户可以进行复制、点赞/点踩、收藏、分享等操作。
- 4. 对于生成的前端代码,用户点击"运行XXX"按钮后可通过底部"添加到工具箱"作为自定 义工具保存,之后在网页端智能问答窗口"我的代码"页面中进行查看或进行再次使用。



支持通过"我的代码"页面进行查看或进行再次使用。

Co	deGeeX					操作手册	
\sim	工作台		SS / 智能问答				
٢	智能问答		Ask CodeGeeX	收藏 我的代码			
ß	工具箱	×	3024				
	个人知识库管理	×	₩ 测试				
	功能配置	~	2024-09-14 19:4	47:28 🔹 查看详情	© 删除代码		
4	下载与支持	×					
	Ξ	_					

我的代码

4.3 工具箱

 在用户后台中,通过工具箱菜单下的默认工具可以使用研发相关的工具,例如代码翻译 等。

	操作手册
88 / 工具箱 / 默认工具	
默认工具 代码翻译 🗸	
输入代码	
请输入要量详的代码	
翻译为 C++	翻译
	■ / 工具種 / 飲以工具 飲以工具 代ி簡請 ● 「 輸入代码 頭協人受助時的代明 」 副能力 C++ ●

4.4 功能配置

4.4.1 配置单元测试框架

在用户后台中,通过在功能配置菜单下的单元测试,可以指定生成单元测试时使用的主流 语言的代码框架。如果全部选择默认,则代码框架同管理员默认配置的一致。

CodeGeeX		操作手册 🌑
🗵 工作台	器 / 功能配置 / 单元测试	
♥ 智能问答	单元测试生成	۷
	▼ Java	
◇ 功能配置 ^	单元测试框架 默认	
单元测试	▼ Python	
三方配置	单元测试框架 默认	
▲ 下载与支持 >	▼ Go	
	单元测试框架 默认	
	▼ JavaScript/TypeScript	
	单元测试框架 默认	
	▼ C/C++	
	单元测试框架 默认	
	L	

4.4.2 三方配置

在用户后台中,通过完成功能配置菜单下的三方配置,可以在客户端中调用第三方的功能。当前支持Gitlab和Gitee。

注意:代码仓库的地址为代码仓库根目录的地址,例 如"http://www.testGitlab.com"。Gitlab的API凭证必须配置api, read_api, read_repository, read_registry的权限,等级最低为maintainer级。

当前支持的Gitlab版本: GitLab Community Edition v16.7.8。

当前支持的Gitee版本: Gitee线上版。

操作手册
۷

4.5 个人知识库管理

4.5.1 个人代码仓库

用户后台支持配置远端私有代码仓库(Git)或本地上传代码仓库,可用于代码仓库级补全功能。在代码仓库级补全功能中,服务端会优先在企业代码库中找到对应的代码项目生成

补全建议,如果管理员没有在管理后台配置对应的企业代码仓库,则会在个人代码仓库中 找到对应的代码项目生成补全建议。

每个用户最多添加10个个人代码仓库,个人代码仓库不支持通过 @repo 进行智能问答, 仅用于代码仓库级补全功能。

当前支持的Gitlab版本: GitLab Community Edition v16.7.8。

当前支持的Gitee版本: Gitee线上版。

- 1. 登录用户后台,在个人代码仓库页面中可以通过添加单个项目或者多个项目的方式添加 代码仓库。
 - 。在个人知识库管理菜单下的个人代码仓库页面,点击新增可添加远端或本地的代码仓
 库。

0	智能问答		11	1 / 个人知识	车管理 / 个人代码仓库	t.								
Ф	数据看板	Ň		仓库名 训制	主人仓库名	项目仓库地址	请输入项目仓库地址	仓库类型	全部	→ 自动索引分支	请输入自动索引分支			重要 查询
0	工具箱	×.												
8	用户管理	~		个人代码仓库	信息 🖌								点击新增	+ 3522
0	日志管理	ř		ID	仓库名	项目仓库地	址	仓	库类型	自动索引分3	र स्र क	更新时间	操作	
6	知识库管理	~												
Φ	个人知识库管理	* ^								\square				
- (个人代码仓库									智无数据				
0	敏感词管理	^												
	词包管理			125	《1×1×1×16世界贝图									
	命中用户													

- 新增代码仓库支持两种方式,分别是添加单个项目和添加多个项目(即整个项目
 组)。添加多个项目组会自动拉取项目组下的所有项目。
 - 添加单个代码仓库时,支持远端和本地两种方式。
 - 本地代码仓库支持通过本地代码项目压缩包进行上传,大小不超过50MB,支持 zip,tar,7z等压缩包类型。通过本地上传添加的代码仓库不支持自动更新,仓 库名称需要和用户IDE中项目名称完全一致才能用于生成代码仓库级补全建议。
 - 远端代码仓库支持Gitlab和Gitee等两种类型的Git代码仓库,可通过通过浏览器
 Url、Git地址、SSH链接等不同入口的链接进行添加。
 - 添加多个代码仓库时, 仅支持远端的方式。
 - 添加Gitlab类型的多个项目时,项目组地址为浏览器中项目组的访问地址。
 - 添加Gitee类型的多个项目时,支持Gitee的地址类型为组织或企业,组织或企 业地址为浏览器中的访问地址。
 - 在Gitee页面位于组织下的项目列表页面时,地址类型为组织(包括Gitee企 业版中组织)。
 - 当使用了Gitee企业版,且Gitee页面为企业下的项目列表页面,地址类型为 企业。

	新增代码仓库		×		新增代码仓	库	
添加方式	添加单个项目	~		添加方式	添加多个项目	~	
仓库类型	Gitlab	~		仓库类型	Gitlab	~	
* 项目仓库地址 🛛	请输入项目仓库地址			* 代码仓库地址 🛛	请输入项目组或代码	马仓库根地址	
* API凭证 🛛	请输入API凭证			* API凭证 ❷	请输入API凭证		
★ 自动索引分支 ❷	请添加自动索引分支 main ×		添加	* 自动索引分支 🛛	请添加自动索引分支 main ×	Z	添加
*自动索引间隔	- 3		小时 +	*自动索引间隔	- 3		小时 +
测试连接			取消 确定	测试连接		Ħ	又消 确定
添加单个项目				添加多个	项目		

- 。 Gitlab的API凭证权限:
 - 添加单个项目时, API凭证需要为项目的凭证或者是项目管理员, 必须配置api, read_api, read_registry的权限, 项目凭证等级最低为reporter级。
 - 添加多个项目时,API凭证需要为项目组的凭证或者是组管理员,必须配置api, read_api, read_user, read_registry, read_repository等权限,项目组凭证等级最 低为reporter级。

U + U	per leftor / Asset Tales B-0 Sent webge Personal Access Tokens	Personal Access Tokens Add a personal access token Teter name For example, the application using the taken or the purpose of the taken.
Eck profile Preferences Sign out	Type can generate a personal access taken for each application you use that weeks access to the GLUb AP. You can also use personal access taken to antheradore applied GL over HTTT. They are the access taken constrained and the personal automatication (CM) existing. Active personal access taken constrained and the personal automatication (CM) existing.	Excitation date Excitation date Excitations Excitations Statest screen
Orde Arcser Stars Orde Arcser Stars Drats Norffectors Drats Dra	Fordiard Control Loss (Mel) Parton fordiardiardia niji madaji madaji madaji madaji maja jimaji	uit common comparer resolver to access to the AdP, blocking algrapse and grapsets, the constance registry, the dependency prose, and the plackage registry. uit_common uit_common <t< th=""></t<>
	找到Gitlab配置页面	配置Gitlab凭证

- 。 Gitee的API凭证权限:
 - 添加单个或多个项目时,必须配置user_info, projects, groups, enterprises的 权限。

and double 111	私人令牌可以使用私人令得访问 Gitee Open API		私人令牌		
M8+0	第試 user_info projects pull_requests issues groups enterprises いた日日		使用私人令辞,可以通过 GRee Open API 话间容极权的数据,请注意致服安全!		
4 10000	↓ 1/110	*=7	私人令牌描述 用于蛇印序管理		
E+43 C E (1) E (1	第二步	E	Light 母生型(加)人(今,40)時間 中以(加)		
	找到Gitee配置	配置Gitee凭证			

当代码仓库的状态为rag同步完成时,用户可以在研发问答中选择对应的代码仓库进行基于RAG的问答。

注意1: 请勿使用不同的地址多次添加同一个代码仓库,数据表中会存为多条记录,但用户 在客户端调用时仅能调用一个。

注意2:本地上传的代码仓库需要填写项目名作为仓库名称,否则本地上传的代码项目无 法用于生成代码仓库级的补全建议。

4.6 可视化编程 / CodeGeeX Reflection

在用户后台中,用户在 可视化编程 页面,可以通过自然语言端到端的生成可运行的前端项 目代码,并通过自然语言或点击页面上的元素进行修改,实时查看代码运行的效果。

器 / 可视化编程		
	将你的创意转化为应用	8
	写一个简单好用UI优美的计算器程序	17/100
1		

生成的前端项目代码会直接在浏览器中运行,可以随时通过自然语言再次修改页面。也可 以选中左侧的放大镜后,悬停在页面的特定元素上,选中该元素进行修改;如果选中成功 后,放大镜会变成确认标志,点击此标志可以取消之前选中的元素。

 回税化编程 可税化编程 育景色修改成渐变色 写一个简单好用UI优美的计算基理序 v1 C 	箱洗中而面的特定元素	再次編編 所会语 第画 平板 平机 � 代码
		通过自然语言修改页面
	计算器	查看页面源码
	7 8 9 C	
	4 5 6 /	
	1 2 3 [•] 0 . = -	
	+	
	结果: 0	

同时,点击右侧的 代码 按钮可以查看该前端项目的源码,并通过修改代码的形式修改页 面元素。生成的前端项目代码可直接下载至本地。

Main.jsx	三方库引用 土 団
<pre>import React, { useState } from 'react'; import ReactDOM from 'react-dom'; const App = () => { const [input, setInput] = useState(''); const [result, setResult] = useState(0); </pre>	查看引用的 三方库
<pre>7 8 const handleButtonClick = (value) => { 9 setInput((prevInput) => prevInput + value); 10 }; 11</pre>	
<pre>const handleClear = () => { setInput(''); setResult(0); }; const handleCalculate = () => { try { setResult(eval(input)); } catch (error) { } }</pre>	下载项目源码
<pre>21 SetResult('Error'); 22 } 23 }; 24 25 return (26 <div classname="min-h-screen bg-gray-100 flex items-center justify-center"> 27 <div classname="min-h-screen bg-gray-100 flex items-center justify-center"> 28 <div classname="min-h-screen bg-gray-100 flex items-center justify-center"> 28 <div classname="min-h-screen bg-gray-100 flex items-center justify-center"> 28 <div classname="min-h-screen bg-gray-100 flex items-center justify-center"> 29 <div classname="min-h-screen bg-gray-100 flex items-center justify-center"> 20 // // // // // // // // // // // // //</div></div></div></div></div></div></pre>	
30 cype=text 31 classMame="w=full p=2 mb=4 border border=gray=300 rounded" 32 value={input} 33 编辑源码	关闭弹窗

5. 使用案例

1. 通过注释及函数定义生成代码:

write a quick sort function in python that use divide and conquer method
def quick_sort(arr):

插件给出的建议(灰色部分),按 Tab 接受建议:

```
# write a quick sort function in python that use divide and conquer method
def quick_sort(arr):
    if len(arr) <= 1:
        return arr
    else:
        pivot = arr[0]
        less_than_pivot = [x for x in arr[1:] if x <= pivot]
        greater_than_pivot = [x for x in arr[1:] if x > pivot]
        return quick_sort(less_than_pivot) + [pivot] + quick_sort(greater_than_pivot) + [pivot] + quick_sort(greater_than_pivot] + [pivot] + quick_
```

注意: 注释中用自然语言描述的部分, 应按照正常编写注释时的叙述方式简单、准确地描述代码段的功能。

不应把它当作ChatGPT类的工具,用对话、指令的方式描述,也不宜提出过于复杂或笼统的任务。

例如,以下是比较好的描述方式:

// 快速排序函数 // Given two non-negative integers, num1 and num2 represented as string, re // read the data in file "abc.txt", then print it out

以下叙述形式CodeGeex可能无法准确理解:

// 写一个python脚本自动抓取GitHub Trending中有关ChatGFT的项目

// 我想用Python爬取豆瓣电影T0P250的数据,代码要怎么写?

// 请写一段javascript代码求解鸡兔同笼问题: 鸡和免一共35个头, 94个脚, 求鸡免分别多少

2. 根据上下文自动生成(灰色部分为CodeGeeX生成):

```
import matplotlib.pyplot as plt
from sklearn import linear_model
import numpy as np
count = 50
noise = 6
x = np.linspace(0, 10, 100)
y = np.sin(x) + np.random.normal(0, noise, 100)
plt.plot(x, y, 'o')
X = x.reshape(-1, 1)
model = linear_model.LinearRegression()
model.fit(X, y)
plt.plot(X, model.predict(X), color='red')
```

3. 智能问答



4. 交互式编程



6. 常见问题解答

如何修改生成注释使用的语言?

答:CodeGeeX的注释生成功能可以生成中文或英文的注释,使用的语言可以在设置中指 定。

- JetBrains IDEAs的设置,我们以 IDEA 为例。如果CodeGeeX在IDEA中生成的注释是英文,按照下面的步骤,在设置中修改为中文即可:设置修改步骤:Settings— Languages & Frameworks—CodeGeeX—Comment Language。
- VSCode 中第一次使用生成注释功能时,IDE会提示选择注释的语言,并让用户选择是 否要设置为默认语言。如果后续需要修改注释的默认语言,可以按照下面的步骤,在 CodeGeeX插件中进行设置: 设置修改步骤:首选项——设置——CodeGeeX—— Coodegeex.Explanation:Language Preference。

如何修改单行/多行的补全方式?

答: 通过以下步骤: 可以设置自己更习惯的补全方式: 单行补全或者多行自动补全。 设置 修改步骤: 首选项——设置——CodeGeeX——Generation Preference。

如何通过设置实现用快捷键触发代码补全功能?

答:

- 在VSCode 中通过以下步骤:勾选后可以控制CodeGeeX 插件,只有在按下"alt/option + 、"快捷键之后才会触发代码补全功能。设置修改步骤:首选项—— 设置——CodeGeeX——Only Key Control。
- 在 JetBrains 的 IDE 中,以 IDEA 为例,如果希望只通过快捷键触发代码补全功能,可以通过以下步骤实现: 右下角CodeGeeX 图标——全局禁用。再进入IDE的设置,指定快捷键来实现按键触发。设置修改步骤: Setting——Keymap——CodeGeeX——Show Completions。

能否用其它快捷键代替"Tab",做为代码生成的采纳键?

答:

- 在VSCode中可以通过设置,将"Tab"键做为代码采纳的快捷键进行修改,更符合自己的使用习惯。修改时要注意,需要把系统默认和 CodeGeeX默认,这 2 个"Tab"键,同时修改才可以生效。设置修改步骤:左下角设置——键盘快捷方式——搜索 "editor.action.inlineSuggest.commit"——修改 2 个默认 Tab 设置。
- 在 JetBrains IDEs中,也可以通过设置,将"Tab"键做为代码采纳的快捷键进行修改, 更符合自己的使用习惯。以 IDEA 为例,通过以下的步骤进行设置修改:设置修改步骤:Setting——Keymap——CodeGeeX——Apply Completions to Editor。

CodeGeeX侧边栏区域无法显示怎么办?

答: 打开 VSCode 后,有时会出现 CodeGeeX侧边栏区域无法显示,或者一直显示加载状态,或者显示"还原视图时出错"、或者显示"无效用户"这样的提示。

解决方法:在MAC电脑的钥匙串访问中,找到当前vscode版本并且尾缀是".codegeex"的项,删除后,重新启动 VSCode,侧边栏就可以正常使用了。

代码库和文档库支持哪些编码格式?

答: CodeGeeX企业版的代码库和文档库支持UTF-8、UTF-16/UTF-32、GBK/GB2312、 ANSI、Windows-1252、ISO-8859-1、Big5、Shift_JIS、EUC-JP、KOI8-R等编码格式。

切换三方模型库后使用问答显示"connection error"怎么办?

答: 联系管理员, 检查对应的三方模型配置。

7.反馈

为了不断提升产品质量和用户体验,我们诚挚邀请您分享CodeGeeX的使用感受和宝贵意见。无论是发现的问题、对功能的期望,还是进一步的需求,都是我们优化和升级的关键 线索,欢迎通过以下渠道分享:

1. 联系企业内部的CodeGeeX系统管理员,并通过系统管理员反馈给企业对接的售后。

2. 反馈问卷:(直达链接或下方扫一扫)



附录

支持的所有编程语言(310种)

```
{'abap': 'ABAP',
    'actionscript': 'ActionScript',
```

```
'ada': 'Ada',
'agda': 'Agda',
'ags_script': 'AGS_Script',
'alloy': 'Alloy',
'ampl': 'AMPL',
'antlr': 'ANTLR',
'apacheconf': 'ApacheConf',
'api_blueprint': 'API_Blueprint',
'apl': 'APL',
'applescript': 'AppleScript',
'arc': 'Arc',
'arduino': 'Arduino',
'aspectj': 'AspectJ',
'assembly': 'Assembly',
'ats': 'ATS',
'augeas': 'Augeas',
'autohotkey': 'AutoHotkey',
'autoit': 'AutoIt',
'awk': 'Awk',
'basic': 'Basic',
'befunge': 'Befunge',
'bison': 'Bison',
'bitbake': 'BitBake',
'blitzbasic': 'BlitzBasic',
'blitzmax': 'BlitzMax',
'bluespec': 'Bluespec',
'boo': 'Boo'.
'brainfuck': 'Brainfuck',
'brightscript': 'Brightscript',
'bro': 'Bro',
'c': 'C'.
'c++': 'C++',
'c2hs_haskell': 'C2hs_Haskell',
"cap'n_proto": "Cap'n_Proto",
'cartocss': 'CartoCSS',
'ceylon': 'Ceylon',
'chapel': 'Chapel',
'chuck': 'ChucK',
'cirru': 'Cirru',
'clarion': 'Clarion',
'clean': 'Clean',
```

```
'click': 'Click',
'clips': 'CLIPS',
'clojure': 'Clojure',
'cmake': 'CMake',
'cobol': 'COBOL',
'coffeescript': 'CoffeeScript',
'coldfusion': 'ColdFusion',
'coldfusion_cfc': 'ColdFusion_CFC',
'common_lisp': 'Common_Lisp',
'component_pascal': 'Component_Pascal',
'coq': 'Coq',
'creole': 'Creole',
'crystal': 'Crystal',
'csharp': 'Csharp',
'csound': 'Csound',
'css': 'CSS',
'cuda': 'Cuda',
'cycript': 'Cycript',
'cython': 'Cython',
<u>'d': 'D',</u>
'darcs_patch': 'Darcs_Patch',
'dart': 'Dart',
'desktop': 'desktop',
'digital_command_language': 'DIGITAL_Command_Language',
'dns_zone': 'DNS_Zone',
'dockerfile': 'Dockerfile',
'dogescript': 'Dogescript',
'dylan': 'Dylan',
'e': 'E',
'ec': 'eC',
'ecere_projects': 'Ecere_Projects',
'ecl': 'ECL',
'edn': 'edn',
'eiffel': 'Eiffel',
'elixir': 'Elixir',
'emacs_lisp': 'Emacs_Lisp',
'emberscript': 'EmberScript',
'erlang': 'Erlang',
'f#': 'F#',
'factor': 'Factor',
'fancy': 'Fancy',
```

```
'fantom': 'Fantom',
'fish': 'fish',
'flux': 'FLUX',
'forth': 'Forth',
'fortran': 'FORTRAN',
'g-code': 'G-code',
'gams': 'GAMS',
'gap': 'GAP',
'genshi': 'Genshi',
'gentoo_ebuild': 'Gentoo_Ebuild',
'gentoo eclass': 'Gentoo Eclass',
'gettext_catalog': 'Gettext_Catalog',
'glyph': 'Glyph',
'go': 'Go',
'golo': 'Golo',
'qosu': 'Gosu',
'grace': 'Grace',
'grammatical_framework': 'Grammatical_Framework',
'graphql': 'GraphQL',
'graphviz': 'Graphviz',
'groff': 'Groff',
'groovy_server_pages': 'Groovy_Server_Pages',
'harbour': 'Harbour',
'haskell': 'Haskell',
'hlsl': 'HLSL',
'html': 'HTML',
'html+django': 'HTML+Django',
'html+eex': 'HTML+EEX',
'html+php': 'HTML+PHP',
'http': 'HTTP',
'hy': 'Hy',
'idl': 'IDL',
'idris': 'Idris',
'igor_pro': 'IGOR_Pro',
'inform_7': 'Inform_7',
'inno_setup': 'Inno_Setup',
'io': 'Io',
'ioke': 'Ioke',
'irc log': 'IRC log',
'isabelle': 'Isabelle',
'i': 'J',
```

```
'jasmin': 'Jasmin',
'java': 'Java',
'javascript': 'JavaScript',
'jflex': 'JFlex',
'json': 'JSON',
'json5': 'JSON5',
'jsoniq': 'JSONiq',
'jsonld': 'JSONLD',
'julia': 'Julia',
'kit': 'Kit',
'kotlin': 'Kotlin',
'krl': 'KRL',
'labview': 'LabVIEW',
'lasso': 'Lasso',
'latte': 'Latte',
'lean': 'Lean',
'lex': 'Lex',
'lfe': 'LFE',
'lilypond': 'LilyPond',
'linker_script': 'Linker_Script',
'liquid': 'Liquid',
'lisp': 'Lisp',
'literate_agda': 'Literate_Agda',
'literate_coffeescript': 'Literate_CoffeeScript',
'literate_haskell': 'Literate_Haskell',
'livescript': 'LiveScript',
'llvm': 'LLVM',
'logos': 'Logos',
'logtalk': 'Logtalk',
'lolcode': 'LOLCODE',
'lookml': 'LookML',
'lsl': 'LSL',
'lua': 'Lua',
'm': 'M',
'm4': 'M4',
'makefile': 'Makefile',
'mako': 'Mako',
'maple': 'Maple',
'markdown': 'Markdown',
'mask': 'Mask',
'mathematica': 'Mathematica',
```

```
'matlab': 'Matlab',
'maxscript': 'MAXScript',
'mediawiki': 'MediaWiki',
'metal': 'Metal',
'minid': 'MiniD',
'mirah': 'Mirah',
'modelica': 'Modelica',
'module_management_system': 'Module_Management_System',
'monkey': 'Monkey',
'moonscript': 'MoonScript',
'mtml': 'MTML',
'muf': 'MUF',
'mupad': 'mupad',
'myghty': 'Myghty',
'nesc': 'nesC',
'netlinx': 'NetLinx',
'netlogo': 'NetLogo',
'nginx': 'Nginx',
'nimrod': 'Nimrod',
'ninja': 'Ninja',
'nit': 'Nit',
'nix': 'Nix',
'nsis': 'NSIS',
'nu': 'Nu',
'objective-c': 'Objective-C',
'objective-c++': 'Objective-C++',
'objective-j': 'Objective-J',
'octave': 'Octave',
'omgrofl': 'Omgrofl',
'ooc': 'ooc',
'opa': 'Opa',
'opal': 'Opal',
'opencl': 'OpenCL',
'openscad': 'OpenSCAD',
'ox': '0x',
'oxygene': '0xygene',
'oz': '0z',
'pan': 'Pan',
'papyrus': 'Papyrus',
'parrot': 'Parrot',
'parrot_assembly': 'Parrot_Assembly',
```

```
'parrot_internal_representation': 'Parrot_Internal_Representation',
'pascal': 'Pascal',
'pawn': 'PAWN',
'perl': 'Perl',
'perl6': 'Perl6',
'php': 'PHP',
'piglatin': 'PigLatin',
'pike': 'Pike',
'pod': 'Pod',
'pogoscript': 'PogoScript',
'pony': 'Pony',
'pov-ray_sdl': 'POV-Ray_SDL',
'powershell': 'PowerShell',
'processing': 'Processing',
'prolog': 'Prolog',
'propeller_spin': 'Propeller_Spin',
'public_key': 'Public_Key',
'pure_data': 'Pure_Data',
'purebasic': 'PureBasic',
'purescript': 'PureScript',
'python': 'Python',
'qmake': 'QMake',
'r': 'R'.
'ragel_in_ruby_host': 'Ragel_in_Ruby_Host',
'raml': 'RAML',
'raw_token_data': 'Raw_token_data',
'rdoc': 'RDoc',
'realbasic': 'REALbasic',
'rebol': 'Rebol',
'red': 'Red',
'redcode': 'Redcode',
'renderscript': 'RenderScript',
'restructuredtext': 'reStructuredText',
'rhtml': 'RHTML',
'rmarkdown': 'RMarkdown',
'robotframework': 'RobotFramework',
'rouge': 'Rouge',
'ruby': 'Ruby',
'rust': 'Rust',
'sage': 'Sage',
'saltstack': 'SaltStack',
```

```
'sas': 'SAS',
'sass': 'Sass',
'scala': 'Scala',
'scaml': 'Scaml',
'scilab': 'Scilab',
'self': 'Self',
'shell': 'Shell',
'shellsession': 'ShellSession',
'shen': 'Shen',
'slash': 'Slash',
'slim': 'Slim',
'smt': 'SMT',
'solidity': 'Solidity',
'sourcepawn': 'SourcePawn',
'sparql': 'SPARQL',
'sqf': 'SQF',
'sql': 'SQL',
'squirrel': 'Squirrel',
'stan': 'Stan',
'standard_ml': 'Standard_ML',
'stata': 'Stata',
'ston': 'STON',
'stylus': 'Stylus',
'supercollider': 'SuperCollider',
'swift': 'Swift',
'systemverilog': 'SystemVerilog',
'tcsh': 'Tcsh',
'tea': 'Tea',
'tex': 'TeX',
'textile': 'Textile',
'thrift': 'Thrift',
'turing': 'Turing',
'txl': 'TXL',
'txt': 'TXT',
'typescript': 'TypeScript',
'unified_parallel_c': 'Unified_Parallel_C',
'uno': 'Uno',
'unrealscript': 'UnrealScript',
'urweb': 'UrWeb',
'vala': 'Vala',
'vcl': 'VCL',
```

```
'verilog': 'Verilog',
'visual_basic': 'Visual_Basic',
'volt': 'Volt',
'vue': 'Vue',
'webassembly': 'WebAssembly',
'webidl': 'WebIDL',
'wisp': 'wisp',
'x10': 'X10',
'xbase': 'xBase',
'xc': 'XC',
'xml': 'XML',
'xojo': 'Xojo',
'xpages': 'XPages',
'xproc': 'XProc',
'xquery': 'XQuery',
'xs': 'XS',
'xtend': 'Xtend',
'yang': 'YANG',
'zephir': 'Zephir',
'zig': 'Zig',
'zimpl': 'Zimpl'}
```